

Альтернативный экзамен по дискретной математике. "Меандры"

Выполнил: Варивода С. А.
Преподаватель: Поздняков С. Н.

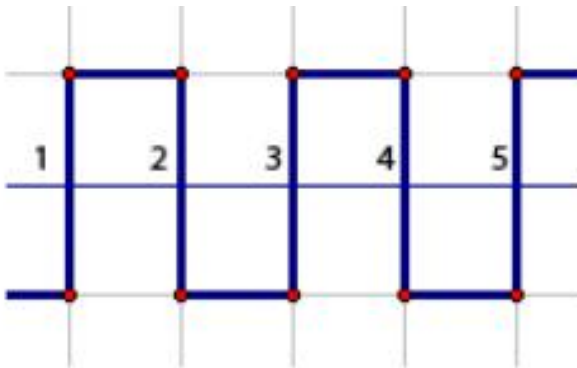
Постановка задач

- Первая задача: дана перестановка - определить, задаёт ли она меандр.
- Вторая задача: сгенерировать все меандры для n пересечений.

Определение меандра

Задача, которую мы будем анализировать, формулируется так:

Шоссе, идущее с запада на восток, пересекает несколько раз реку, текущую с юго-запада на восток. Занумеруем мосты в порядке их следования вдоль шоссе (с запада на восток). Проплывая под мостами вниз по реке, мы будем встречать их, вообще говоря, в другом порядке. Например, 3,4,5,2,1. Таким образом, эта река определяет перестановку. Другая река могла бы задавать другую перестановку. Но далеко не любая перестановка чисел (мостов) может быть реализована таким образом. Например, не придумать реку, проходящую через мосты в порядке 2,1,3,4,5. Будем называть перестановку меандром, если ее можно задать с помощью подходящей реки.

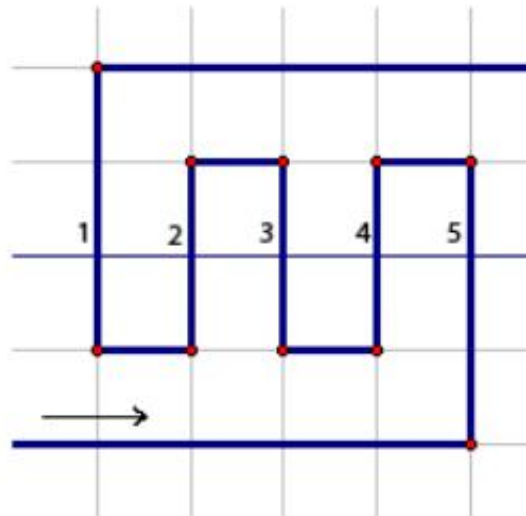


Решение первой задачи

Проведя исследование, я заметил, что меандр должен соответствовать двум условиям:

- 1) Начинаться с нечетного моста
- 2) Дальше четные и нечетные мосты должны чередоваться

То есть четность позиции (если считать с 1) должна соответствовать четности номера моста.



Решение первой задачи

Но оговоренных выше условий недостаточно, поэтому я предлагаю следующий алгоритм проверки меандра.

- 1) Создадим переменную-флаг, которая отмечает "снизу" или "сверху" мы пришли к i -ому элементу.
- 2) Создадим два массива, верхних и нижних ограниченных отрезков.
- 3) На каждой итерации записываем в соответствующий массив "заблокированный отрезок"
- 4) Смотрим, попадает ли элемент в "заблокированный отрезок".
- 5) Если попадает, то проверяем предыдущий элемент, и если он не в отрезке, то это пересечение.

Решение первой задачи

Рассмотрим пример.

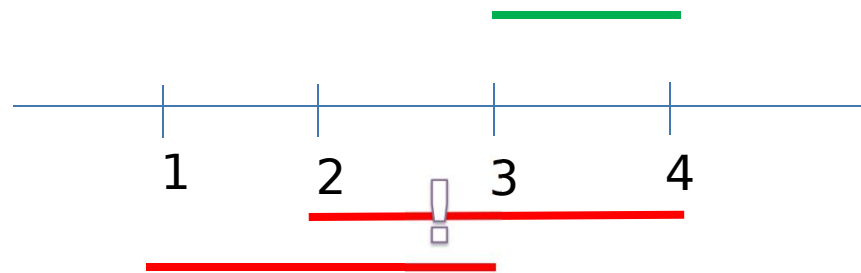
Перестановка: 3 4 2 1

Левая граница: 1

Флаг: снизу

Массив нижних отрезков (Н): пустой

Массив верхних отрезков (М): пустой



1) Т.к. массивы пустые записываем в Н (т.к. флаг снизу) [1, 3], меняем флаг на "вверху" и левую границу на 3.

2) 4 не попадает в отрезки, поэтому просто добавляем в М [3,4], меняем флаг на "снизу" и левую границу на 4.]

3) К 2 мы пришли снизу и попадаем в отрезок Н - [1,3].
Значит смотрим на предыдущий элемент. 4 вне этого отрезка.

Это не меандр.

Решение второй задачи

Воспользуемся стандартным алгоритмом генерации лексикографических перестановок без повторений.

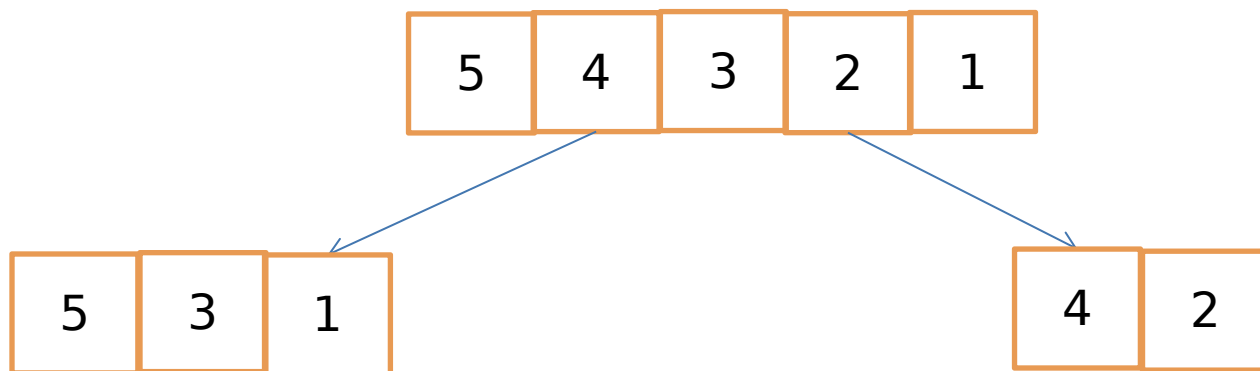
Дана исходная последовательность чисел. Для получения каждой следующей перестановки необходимо выполнить следующие шаги:

- 1) Необходимо просмотреть текущую перестановку справа налево и при этом следить за тем, чтобы каждый следующий элемент перестановки (элемент с большим номером) был не более чем предыдущий (элемент с меньшим номером).
- 2) Как только данное соотношение будет нарушено необходимо остановиться и отметить текущее число (позиция 1).
- 3) Снова просмотреть пройденный путь справа налево пока не дойдем до первого числа, которое больше чем отмеченное на предыдущем шаге.
- 4) Поменять местами два полученных элемента.
- 5) Теперь в части массива, которая размещена справа от позиции 1 надо отсортировать все числа в порядке возрастания. Поскольку до этого они все были уже записаны в порядке убывания необходимо эту часть подпоследовательность просто перевернуть.

Решение второй задачи

Сложность данного алгоритма $O(n!)$.

Чтобы немного ускорить его я разделил перестановку на 2 массива четных и нечетных чисел.



И в двойном цикле мы делаем перестановки отдельно для каждого массива. После чего собираем обратно в один массив и облегченной проверкой проверяем на пересечения.

Используемые технологии

В качестве основного языка программирования выбран Java.

Выбор был сделан опираясь на главное преимущество этого языка - мультиплатформенность.

